Problem A
Black and White
Time Limit: 1 Second(s)

Black and White is a Chinese children's game played in rounds. During each round, the children who are playing all put their hands in either face-up ("White") or face-down ("Black"). If all the children but one make the same choice, then the "odd one out" sits out for the rest of the game. Play continues until there are only two children left.

Each child independently chooses whether to put their hand face-up with their own fixed probability. What is the expected number of rounds that such a game will last?

Input

The first line contains a single integer n ($2 \leq n \leq 20$), which is the number of children.

Each of the next n lines contains a single real number p ($0.1 \leq p \leq 0.9$). These are the probabilities for each child that they will put their hand in face-up. The probabilities will have at most three digits after the decimal point.

Output

Output a single real number, which is the expected number of rounds. The result must be accurate to within an absolute or relative error of 10^{-6}.

Sample Input 1	Sample Output 1
3 0.5 0.5 0.5	1.3333333

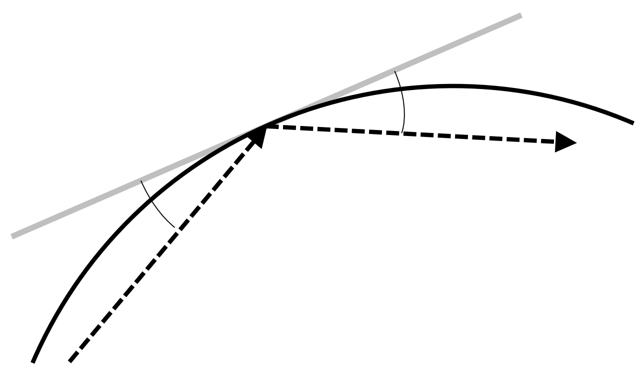
Sample Input 2	Sample Output 2
3 0.3 0.3 0.3	1.5873015

Sample Input 3	Sample Output 3
5 0.1 0.3 0.5 0.7 0.9	7.4752846

Problem B
Circle Bounce
Time Limit: 1 Second(s)

You are standing by the wall in a large, perfectly circular arena and you throw a tennis ball hard against some other part of the arena. After a given number of bounces, where does the tennis ball next strike the wall?

Map the arena as a unit circle centered at the origin, with you standing at the point $(-1, 0)$. You throw the ball with a direction given by a slope in the coordinate plane of a rational fraction a/b. Each bounce is perfect, losing no energy and bouncing from the wall with the same angle of reflection as the angle of incidence to a tangent to the wall at the point of impact.



After n bounces, the ball strikes the circle again at some point p which has rational coordinates that can be expressed as $(r/s, t/u)$. Output the fraction r/s modulo the prime $M = 1{,}000{,}000{,}007$.

It can be shown that the x coordinate can be expressed as an irreducible fraction r/s, where r and s are integers and $s \not\equiv 0 \pmod{M}$. Output the integer equal to $r \cdot s^{-1} \pmod{M}$. In other words, output an integer k such that $0 \le k < M$ and $k \cdot s \equiv r \pmod{M}$.

For example, if we throw the ball with slope $1/2$ and it bounces once, it first strikes the wall at coordinates $(3/5, 4/5)$. After bouncing, it next strikes the wall at coordinates $(7/25, -24/25)$. The modular inverse of 25 with respect to the prime M is 280,000,002, and the final result is thus $7 \cdot 280{,}000{,}002 \pmod{M} = 960{,}000{,}007$.

Input

The single line of input will contain three integers a, b ($1 \le a, b \le 10^9, \gcd(a, b) = 1$) and n ($1 \le n \le 10^{12}$), where a/b is the slope of your throw, and n is the number of bounces. Note that a and b are relatively prime.

Output

Output a single integer value as described above.

- Note that Sample 2 corresponds to the example in the problem description.

Sample Input 1	Sample Output 1
`1 1 3`	`1000000006`

Sample Input 2	Sample Output 2
`1 2 1`	`960000007`

Sample Input 3	Sample Output 3
`11 63 44`	`22`

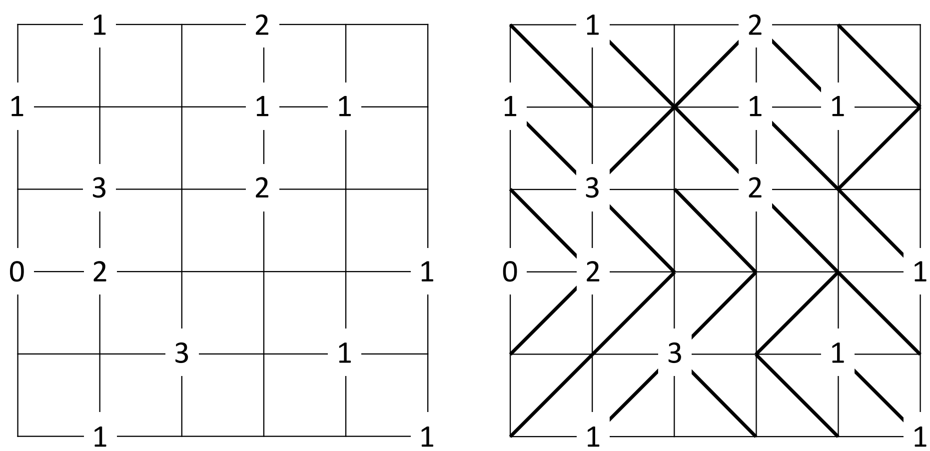
Sample Input 4	Sample Output 4
`163 713 980`	`0`

Problem C
Diagonals
Time Limit: 10 Second(s)

Diagonals is a pencil puzzle which is played on a square grid. The player must draw a diagonal line corner to corner in every cell in the grid, either top left to bottom right, or bottom left to top right. There are two constraints:

- Some intersections of gridlines have a number from 0 to 4 inclusive on them, which is the exact number of diagonals that must touch that point.
- No set of diagonals may form a loop of any size or shape.

The following is a 5×5 example, with its unique solution:



Given the numbers at the intersections of a grid, solve the puzzle.

Input

The first line of input contains an integer n ($1 \leq n \leq 8$), which is the size of the grid.

Each of the next $n+1$ lines contains a string s ($|s| = n+1, s \in \{0, 1, 2, 3, 4, +\}^*$). These are the intersections of the grid, with `'+'` indicating that there is no number at that intersection.

The input data will be such that the puzzle has exactly one solution.

Output

Output exactly n lines, each with exactly n characters, representing the solution to the puzzle. Each character must be either `'/'` or `'\'`.

- Note that Sample 1 corresponds to the example in the problem description.

Sample Input 1

```
5
+1+2++
1++11+
+3+2++
02+++1
++3+1+
+1+++1
```

Sample Output 1

```
\\/\\
\/\\/
\\\\\
////\
//\\\
```

Sample Input 2

```
3
++++
+1+1
+31+
+0+0
```

Sample Output 2

```
/\/
///
/\/
```

Sample Input 3

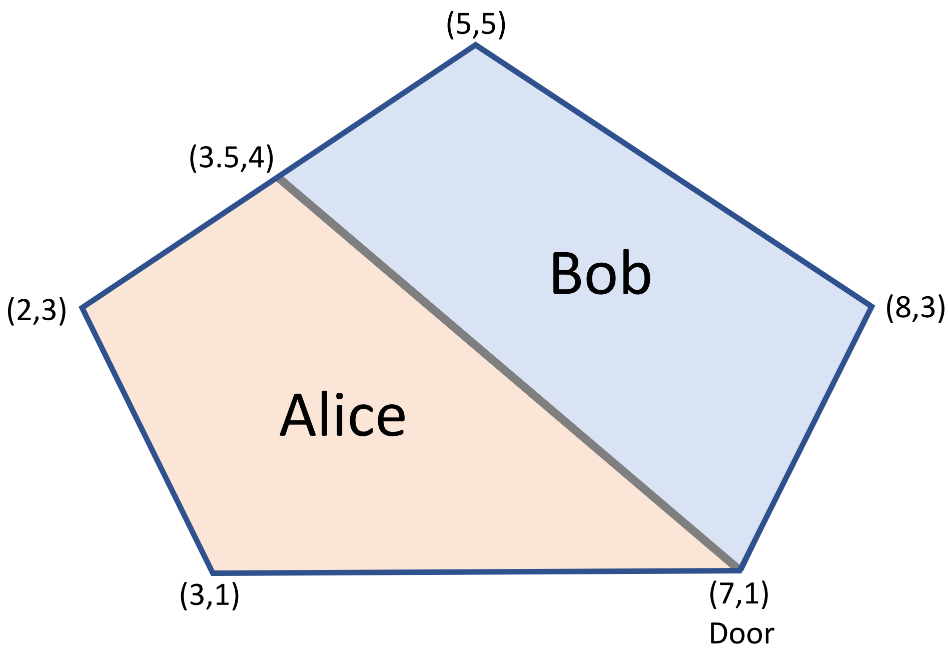
```
4
+++++
+3++2
++3++
+3+3+
++2+0
```

Sample Output 3

```
\//\
\\//
\\\/
/\//
```

Problem D
Dorm Room Divide
Time Limit: 1 Second(s)

Bob and Alice are roommates at the International College of Polygonal Chambers (ICPC). To avoid conflict, they've agreed to divide their dorm room in half—as closely as possible. However, the room is shaped so irregularly that they need your help!



Each dorm room is a convex polygon, with a single entrance. You need to figure out how to divide this room in half (by area) using a single straight line starting at the door, and terminating on a wall or corner of the room.

Input

The first line of input contains a single integer n ($3 \leq n \leq 2 \cdot 10^5$), which is the number of vertices describing the convex polygon.

Each of the next n lines contains two space-separated integers x and y ($-10^7 \leq x, y \leq 10^7$). These are the coordinates of the vertices of the convex polygon, in counterclockwise order. All points will be distinct.

The door is considered to be a single point located at the first vertex given in the input.

Output

Output two space-separated real numbers, which are the x and y coordinates of the other endpoint of the dividing line, such that the area of the room is divided in half. Each coordinate value must be accurate to within an absolute or relative error of 10^{-6}. Output x first, then y.

- Note that Sample 1 corresponds to the example in the problem description.

Sample Input 1	Sample Output 1
5 7 1 8 3 5 5 2 3 3 1	3.5 4

Sample Input 2	Sample Output 2
3 2 2 10 3 6 8	8 5.5

Problem E
Fail Them All!
Time Limit: 3 Second(s)

You are an instructor for an algorithms course, and your students have been saying mean things about you on social media. Those jerks! Being a vengeful and dishonest instructor, you are going to make them pay.

You have given your students a True/False exam. For each question, each student is allowed to either answer the question or leave the question blank. Each student has answered at least two questions. You want to make sure that every student fails the test, so you are going to alter the answer key so that no student gets more than one answer correct.

Is there an answer key such that every person has at most one submitted answer that is correct? If so, compute the lexicographically minimal such answer key.

Input

The first line of input contains two integers n $(1 \leq n \leq 100)$ and k $(2 \leq k \leq 100)$, where n is the number of students in the class, and k is the number of questions on the test.

Each of the next n lines contains a string s $(|s| = k, s \in \{\texttt{T}, \texttt{F}, \texttt{X}\}^*)$, which are the answers to the questions, in order, for each student, where `'T'` means True, `'F'` means False, and `'X'` means the student didn't answer the question. Every student's answers will have at least two which are not `'X'`.

Output

If such an answer key can be constructed, output a string of length k consisting of only the characters `'T'` and `'F'`, which is the answer key. If more than one such key is possible, output the one which comes first alphabetically (`'F'` < `'T'`). If no such key exists, instead output `-1`.

Sample Input 1	Sample Output 1
3 3 FFX XFF FXF	FTT

Sample Input 2	Sample Output 2

`3 3`
`FTX`
`XFT`
`TXF`
	`FFF`

Sample Input 3	Sample Output 3

`4 3`
`TTX`
`XTT`
`TXT`
`FFF`
	`-1`

Problem F
Hopscotch 500
Time Limit: 2 Second(s)

Do you remember the new art installation from NAC 2020? Well, that artist is at it again, on a grander scale this time, and the new artwork still inspires you—to play a childish game. The art installation consists of a floor with a square matrix of tiles. Each tile holds a single number from 1 to k.

You want to play hopscotch on it! You want to start on some tile numbered 1, then hop to a tile numbered 2, then 3, and so on, until you reach a tile numbered k.

Instead of the usual Euclidean distance, define the distance between the tile at (x_1, y_1) and the tile at (x_2, y_2) as:

$$\min\left[(x_1 - x_2)^2, (y_1 - y_2)^2\right]$$

You want to hop the shortest total distance overall, using this new distance metric. Note that a path with no hops is still a path, and has length 0. What is the length of the shortest path?

Input

The first line of input contains two space-separated integers n ($1 \le n \le 500$) and k ($1 \le k \le n^2$), where the art installation consists of an $n \times n$ matrix with tiles having numbers from 1 to k.

Each of the next n lines contains n space-separated integers x ($1 \le x \le k$). These are the numbers in the art installation.

Output

Output a single integer, which is the total length of the shortest path from any 1 tile to any k tile using our distance metric, or -1 if no such path exists.

Sample Input 1

```
10 5
5 1 3 4 2 4 2 1 2 1
4 5 3 4 1 5 3 1 1 4
4 2 4 1 5 4 5 2 4 1
5 2 1 5 5 3 5 2 3 2
5 5 2 3 2 3 1 5 5 5
3 4 2 4 2 2 4 4 2 3
1 5 1 1 2 5 4 1 5 3
2 2 4 1 2 5 1 4 3 5
5 3 2 1 4 3 5 2 3 1
3 4 2 5 2 5 3 4 4 2
```

Sample Output 1

```
0
```

Sample Input 2

```
10 30
18 13 30 15 18 16 14 1 5 5
17 18 7 30 14 30 13 14 1 28
28 24 7 23 9 10 5 12 21 6
11 16 6 2 27 14 1 26 7 21
16 2 9 26 6 24 22 12 8 16
17 28 29 19 4 6 21 19 6 22
11 27 11 26 13 23 10 3 18 6
14 19 9 8 17 6 16 22 24 1
12 19 10 21 1 8 20 24 29 21
21 29 1 23 23 24 6 20 25 17
```

Sample Output 2

```
19
```

Problem G
Shortest Missing Subsequences
Time Limit: 8 Second(s)

Given a string s we say that string t is a *Subsequence* of s if t can be obtained from s by deleting zero or more characters of s. Note that t is not necessarily a substring of s—that is, t is not necessarily contiguous in s, but the characters of t appear in the same order as they do in s.

For a given subset, v, of the lowercase English alphabet characters from `'a'` to `'z'`, we say that string u is a *Missing Subsequence* of another string s if u is not a *Subsequence* of s, but all characters in u and all the characters of s are in the set v. A *Shortest Missing Subsequence* of s is a *Missing Subsequence* of s with the smallest length among all *Missing Subsequences* of s.

Given a set of English alphabetic characters, a target string made up of characters from that set, and a list of query strings made up of characters from that set, determine if each of the query strings is a *Shortest Missing Subsequence* of the target string.

Input

The first line of input contains a string v ($1 \leq |v| \leq 26$) of lowercase letters, in lexicographical order. Each letter appears at most once. This is the set of alphabetic characters.

The next line of input contains a string s ($1 \leq |s| \leq 10^6$, s only contains letters from v). This is the target string to be queried.

The next line contains an integer n ($1 \leq n \leq 10^6$). This is the number of queries.

Each of the next n lines contains a string q ($1 \leq |q| \leq 10^6$, q only contains letters from v). These are the query strings. The sum of the lengths of all query strings will not exceed 10^6.

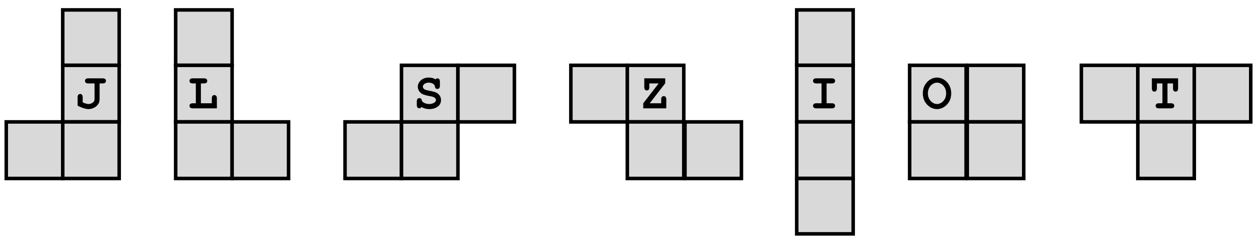
Output

Output n lines, one for each query. On each line, output either 1 if the query string is a *Shortest Missing Subsequence* of the target string, or 0 if it is not. The outputs must be in the order of the input queries.

Sample Input 1	Sample Output 1
abc abcccabac 3 cbb cbba cba	1 0 0

Problem H
Tetris Generation
Time Limit: 1 Second(s)

The classic game Tetris involves arranging falling tetrominoes on a board. There are seven different tetrominoes, each named after a letter that resembles their shape: `J`, `L`, `S`, `Z`, `I`, `O`, and `T`.



In the original Tetris, the player would receive one tetromino at a time, and each tetromino would be chosen from among the seven possibilities independently and uniformly at random. This meant that any sequence of tetrominoes could appear in a game, such as numerous `I` tetrominoes in a row. Modern versions of Tetris remove these streaks by generating tetrominoes in groups of seven: The first seven tetrominoes in a game will be one of each of the seven different tetrominoes in a random order. The next seven tetrominoes will also be one of each of the seven different tetrominoes in a random order (possibly but not necessarily different from the ordering of the first seven). Same goes for the next seven, and so on and so forth. With this generator, it is still possible to get two of the same tetromino in a row (for example, the seventh and eighth tetrominoes in the game can be the same as each other), but it is not possible to get three of the same type in a row.

Given a sequence of tetrominoes, determine whether it is possible for a modern Tetris generator to produce that sequence at some point in a game.

Input

The first line of input contains an integer t ($1 \le t \le 10^5$), which is the number of test cases.

Each of the next t lines contains a single string s ($1 \le |s| \le 1{,}000$, $s \in \{\texttt{J}, \texttt{L}, \texttt{S}, \texttt{Z}, \texttt{I}, \texttt{O}, \texttt{T}\}^*$). This string represents a sequence of tetrominoes, and is a single test case.

The sum of the lengths of all input test cases will not exceed 10^5.

Output

For each test case, output a single line with a single integer, which is `1` if the sequence can be generated by a modern Tetris generator, and `0` otherwise.

Sample Input 1	Sample Output 1
2 JJTO JJTT	1 0

Problem I
Tournament Seeding
Time Limit: 2 Second(s)

You are tasked with seeding a single-elimination tournament for a one-on-one game. The number of players who have registered for the tournament is exactly a power of two, and there will be exactly enough rounds in this tournament to decide a winner. Furthermore, each player has a unique numeric rating in the game known to you; when two players play against each other in a game, the player with the higher rating always wins. As the organizer of the tournament, you would like to make the tournament as exciting for players and spectators as possible. To do that, you wish the tournament to have the following properties:

- The top two (highest rated) players are present in the final round of the tournament, the top four players are present in the semi-final round of the tournament, the top eight players are present in the quarter-final round, and so on. This saves the highest rated games for last.
- Subject to the above, as many games as possible are "close." We define a game to be "close" if the difference between the two players' ratings is less than or equal to some threshold.

Given the number of rounds, the threshold for "close" games and the ratings of the players, what is the maximum number of "close" games that can happen subject to the above constraints?

Input

The first line of input contains two integers n ($1 \le n \le 18$) and k ($1 \le k \le 10^9$), where n is the number of rounds of the tournament, and k is the rating difference that makes a game "close."

Each of the next 2^n lines contains a single integer r ($1 \le r \le 10^9$) denoting the rating of each player. The ratings are guaranteed to be distinct.

Output

Output a single line with a single integer, which is the maximum number of "close" games possible in a tournament among these players satisfying the constraints described above.

Sample Input 1	Sample Output 1
2 2 9 1 6 4	1

Sample Input 2	Sample Output 2
2 5 9 1 6 4	3

Problem J
Tree Hopping
Time Limit: 1 Second(s)

You are given a tree and a permutation of its vertices. It can be proven that for any tree and any pair of source/destination nodes, there is some permutation of the nodes where the first node is the source, the last node is the destination, and the distance between adjacent nodes in the permutation is less than or equal to three.

Your job will be to write a verifier for this property. Given such a permutation and the tree, validate whether the distance between adjacent nodes in the permutation is less than or equal to three.

Input

The first line of input contains an integer t ($1 \le t \le 50{,}000$), which is the number of test cases.

In each test case, the first line of input contains an integer n ($2 \le n \le 100{,}000$), which is the number of nodes in the tree. The nodes are numbered from 1 to n.

Each of the next $n - 1$ lines contains a pair of integers a and b ($1 \le a < b \le n$), representing an edge in the tree between nodes a and b.

Each of the next n lines contains an integer p ($1 \le p \le n$, all values distinct). This is the permutation of the nodes.

The sum of the values of n over all test cases will not exceed $100{,}000$.

Output

For each test case, output a single line with a single integer, which is 1 if the given permutation satisfies the constraint that every pair of adjacent nodes in the permutation has distance less than or equal to three in the tree. Output 0 if the given permutation does not satisfy this constraint.

Sample Input 1	Sample Output 1
2 5 1 2 2 3 3 4 4 5 1 3 2 5 4 5 1 2 2 3 3 4 4 5 1 5 2 3 4	1 0

Problem K
XOR Island
Time Limit: 5 Second(s)

On an island populated entirely with perfect logicians, each islander is wearing a hat that displays a positive integer. Each islander can see all other islanders' hats, but they cannot see their own hat. An islander has no information about the number on their own hat, other than the fact that it is a positive integer.

One day, a mysterious message appears in the sky and says "There exist three distinct islanders such that the XOR of the integers on two of their hats is the integer on the hat of the third". After this message appears, the islanders schedule meetings for several days in a row, one meeting per day. At each meeting any islander who knows for sure that they are part of some triple that satisfies the message will raise their hand.

Assuming that no islanders lie or make mistakes, and each will raise their hand as soon as it is possible for them to know that they are part of a triple, how many days will it take for at least one islander to raise their hand at a meeting?

Input

The first line of input contains a single integer n ($3 \leq n \leq 25$), the number of islanders.

Each of the next n lines contains a single integer a ($1 \leq a < 2^{25}$), which are the positive integers on the islanders' hats.

It is guaranteed that the input is chosen such that the message in the sky is true; there is at least one triple among the islanders' hats such that one hat's integer is equal to the XOR of the integers on the other two hats.

Output

Output a single integer, which is the number of daily meetings it will take for some islander to figure out that their hat is part of some XOR triple. It can be proven that at least one person will eventually raise their hand.

Sample Input 1	Sample Output 1
3 1 2 3	1

Sample Input 2	Sample Output 2
11 9 1 14 2 11 7 6 7 6 5 3	3